Antonina Nazarova

BISC481

Third Assignment

Modeling of protein-DNA binding specificity/ Statistical machine learning.

**1.**

a)Done

b)Done

c) I pushed my report and R script to the repository.

**2.**

Nowadays machine learning plays a key role in DNA/protein binding specificity predictions. There are two types of ways for building the models : for in vivo and in vitro experiments.

Thought these experiments have a lot of difference they all have the same structure for model development:

First, we should have enough valid data in our training data set to allow on.

Second, we should develop an accurate algorithm, that will also minimize cost function.

Third, by inputting our proved data to our algorithm we should be able to build a model.

Fourth and final, by introducing any data testing dataset we should be able to end up with a correct prediction for our analyzing data.

DNA shaper, as well as Biocondutor packages are using for DNA shape prediction and encoding by applying machine learning R package Caret.

a)In vitro experiments SELEX-seq and PBM applies linear regression as the fundamental math principle, different types of accessing the model and the coefficient of determination- R2, as the tool for comparing model results. As we are dealing with the big data (protein structure) we should be able to assign every protein special number based on shape and sequence features , so that at the end we have final feature vector consisting only from the numbers. For example, (Adenine- 0001, Cytozine-0010, Guanine- 0100, Thymine- 1000). While building a new model we should try to minimize cost function for every hypothesis one. Basically we should try to make our paraboloid cost function more narrow and aiming to Y=0. On this stage we use R studio statistical computing use gradient descent algorithm to prove or disprove our models. Then we use different ways for accessing linear regression models:

- random subsampling ( in each experiment test example is randomly distributed among the total number of examples in one experiment);

- k-fold cross validation ( in each experiment test examples are ordered distributed among the total number of examples within one example);

- LOOCV (there is only one single test example that could be found randomly in experiment N)

Overall our coefficient of determination that demonstrates efficiency of satisfaction of our data with a statistical model R2 should be aimed to 1.

b) In vivo experiment ChIP-seq applies logistic regression as the fundamental math principle, different types of accessing the model and the area under ROC curve- AUC, as the tool for comparing model results. Logistic regression or classification divides our data onto two classes by assigning them “0”-number in case of negative class and “1”- number in case of positive class. Cost function has log dependence and hypothesis function-reverse exponential, that is why the closer to 1 is AUC the less overlap between true positive and False positive functions.

c)

3.

a) R was installed;

b) Bioconductor was installed;

c) package DNAshapeR was installed;

d) machine learning package caret was installed;

e) gcPBM in vitro experimental data was downloaded.

The directory was established

workingPath <- "/Users/antonina/Downloads/BISC481-master/gcPBM/"

**4.**

a)

**Here I was changing Myc.txt.fa on Mad.txt.fa and Myc.txt.fa** ## Predict DNA shapes

fn\_fasta <- paste0(workingPath, "Myc.txt.fa")

pred <- getShape(fn\_fasta)

**Here I was changing 1-mer on 1-mer , 1-shape.**

## Encode feature vectors

featureType <- c("1-mer", "1-shape")

featureVector <- encodeSeqShape(fn\_fasta, pred, featureType)

head(featureVector)

**Here I was changing Myc.txt.fa on Mad.txt.fa and Myc.txt.fa**

## Build MLR model by using Caret

# Data preparation

fn\_exp <- paste0(workingPath, "Myc.txt")

exp\_data <- read.table(fn\_exp)

df <- data.frame(affinity=exp\_data$V2, featureVector)

# Arguments setting for Caret

trainControl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)

# Prediction without L2-regularized

model <- train (affinity~ ., data = df, trControl=trainControl,

method = "lm", preProcess=NULL)

summary(model)

# Prediction with L2-regularized

model2 <- train(affinity~., data = df, trControl=trainControl,

method = "glmnet", tuneGrid = data.frame(alpha = 0, lambda = c(2^c(-15:15))))

model2

result <- model2$results$Rsquared[1]

head(result)

**Overall I ended up with 6 numbers of R2:**

**1-mer**

**Mad 0,775645894510131**

**Max 0,785725241119817**

**Myc 0,777694350090362**

**1-mer and 1-shape**

**Mad 0,86366603669467**

**Max 0,864378551220418**

**Myc 0,854713629151327**

b)

######################################

# 11.10.2016

# Multiple Linear Regression (MLR) example

# BISC 481

######################################

## Install and initialize packages

install.packages("ggplot2")

install.packages("grid")

library(ggplot2)

library(grid)

workingPath <- "/Users/antonina/Downloads/BISC481-master/gcPBM/"

## Theme

my.theme <- theme(

plot.margin = unit(c(0.1, 0.5, 0.1, 0.1), "cm"),

axis.text = element\_text(colour="black", size=12),

axis.title.x = element\_text(colour="black", size=12),

axis.title.y = element\_text(colour="black", size=12),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

axis.line = element\_line(colour = "black"),

axis.text = element\_text(colour ="black"),

axis.ticks = element\_line(colour = "black")

)

## Data preparation

**Here I inserted for data 1 tree numbers for feature vectors for “1-mer” sequence model for the datasets of Mad, Max and Myc.**

data1 <- c(0.775645894510131, 0.785725241119817, 0.777694350090362)

**Here I inserted for data 2 tree numbers for feature vectors for “1-mer”, “1-shape” sequence model for the datasets of Mad, Max and Myc.**

data2 <- c(0.86366603669467, 0.864378551220418, 0.854713629151327)

## Ploting

ggplot() +

geom\_point(aes(x = data1, y = data2), color = "red", size=1) +

geom\_abline(slope=1) + geom\_vline(xintercept=0) + geom\_hline(yintercept=0) +

coord\_fixed(ratio = 1, xlim = c(0,1), ylim = c(0,1)) +

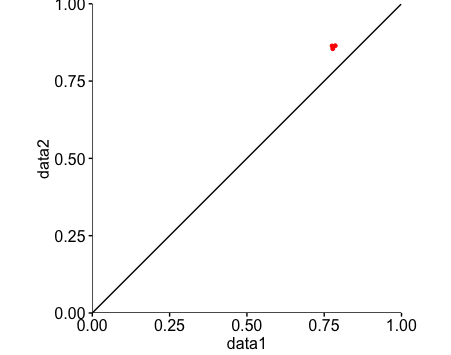
scale\_x\_continuous(expand = c(0, 0)) + scale\_y\_continuous(expand = c(0, 0)) +

my.theme

**My R2= 0,775645894510131**

**5.**

a) b) A plot of comparison F( 1-mer+shape)= x\* F (1-mer) demonstrates that in our protein dataset that is shape that provide additional usefull information for DNA binding specificity prediction, as out predicted R2>0.5.



**6.**

a) ChIP-seq data was downloaded;

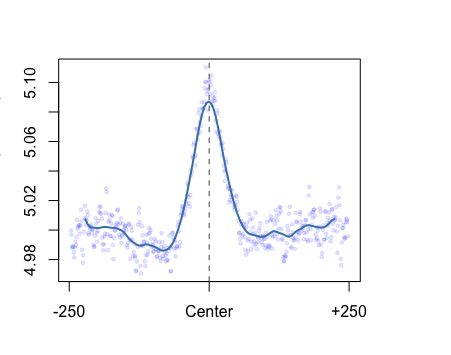
b) R packages were installed.

7.

a)Using plotShape() and heatShape() functions one is able to see which base pairs participate in binding and which are not. The plots have different shape, that is because in plotShape() function demonstrate us the binding base pairs that are under “Gaussian” peak, wherereas heatShape() function demonstrate the same binding base pairs in the different way- by varying the number of base pairs for averaging amoung one bite.

plotShape()

MGW



ProT

/Users/antonina/Desktop/VISA/ProT.pdf

Roll

/Users/antonina/Desktop/VISA/Roll.pdf

HelT

/Users/antonina/Desktop/VISA/plotshapeHelT.pdf

HeatShape

MGW, 10 base pairs

/Users/antonina/Desktop/VISA/MGW_10.pdf

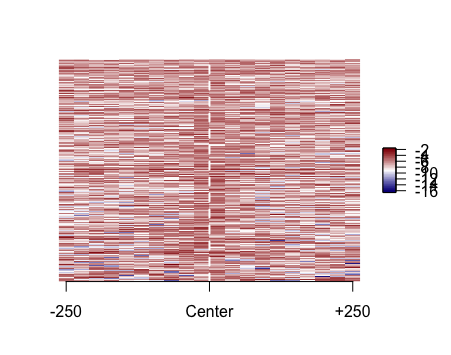
MGW, 20 base pairs

/Users/antonina/Desktop/VISA/MGW_20.pdf

ProT, 10 base pairs

/Users/antonina/Desktop/VISA/ProT_10.pdf

ProT, 20 base pairs



######################################

# 01.10.2016

# Emsemble plots example

# BISC 481

######################################

# Initialization

library(DNAshapeR)

# Extract sample sequences

fn <- "/Users/antonina/Downloads/BISC481-master/CTCF/bound\_500.fa"

# Predict DNA shapes

pred <- getShape(fn)

**Here I was changing MGW on ProT, Roll and HellT.**

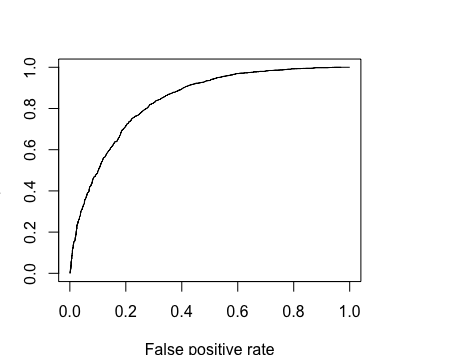
**Then I was changing plotShape() on heatShape(pred$..., 10 or 20)**

# Generate ensemble plots

plotShape(pred$MGW)

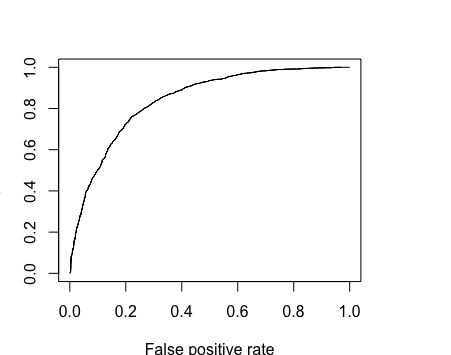
**8.**

a) Logistic regression model for “1-mer” :



AUC= 0,8406546

Logistic regression model for “1-mer+shape”:



AUC= 0,8411313

######################################

# 12.10.2016

# Logistic regression on ChIP-seq data

# BISC 481

######################################

## Install packages

install.packages("caret")

install.packages("e1071")

install.packages("ROCR")

biocLite("Biostrings")

## Initialization

library(DNAshapeR)

library(caret)

library(ROCR)

source("https://bioconductor.org/biocLite.R")

biocLite("Biostrings")

library(Biostrings)

workingPath <- "/Users/antonina/Downloads/BISC481-master/CTCF/"

## Generate data for the classifcation (assign Y to bound and N to non-bound)

# bound

boundFasta <- readDNAStringSet(paste0(workingPath, "bound\_30.fa"))

sequences <- paste(boundFasta)

boundTxt <- data.frame(seq=sequences, isBound="Y")

# non-bound

nonboundFasta <- readDNAStringSet(paste0(workingPath, "unbound\_30.fa"))

sequences <- paste(nonboundFasta)

nonboundTxt <- data.frame(seq=sequences, isBound="N")

# merge two datasets

writeXStringSet( c(boundFasta, nonboundFasta), paste0(workingPath, "ctcf.fa"))

exp\_data <- rbind(boundTxt, nonboundTxt)

## DNAshapeR prediction

pred <- getShape(paste0(workingPath, "ctcf.fa"))

## Encode feature vectors

**Here I was changing “1-mer” on “1mer”, “1-shape”**

featureType <- c("1-mer")

featureVector <- encodeSeqShape(paste0(workingPath, "ctcf.fa"), pred, featureType)

df <- data.frame(isBound = exp\_data$isBound, featureVector)

## Logistic regression

# Set parameters for Caret

trainControl <- trainControl(method = "cv", number = 10,

savePredictions = TRUE, classProbs = TRUE)

# Perform prediction

model <- train(isBound~ ., data = df, trControl = trainControl,

method = "glm", family = binomial, metric ="ROC")

summary(model)

## Plot AUROC

prediction <- prediction( model$pred$Y, model$pred$obs )

performance <- performance( prediction, "tpr", "fpr" )

plot(performance)

## Caluculate AUROC

auc <- performance(prediction, "auc")

auc <- unlist(slot(auc, "y.values"))

auc

## Encode feature vectors

**Here I was changing “1-mer” on “1-mer”, “1-shape”**

featureType <- c("1-mer")

featureVector <- encodeSeqShape(paste0(workingPath, "ctcf.fa"), pred, featureType)

df <- data.frame(isBound = exp\_data$isBound, featureVector)

## Logistic regression

# Set parameters for Caret

trainControl <- trainControl(method = "cv", number = 10,

savePredictions = TRUE, classProbs = TRUE)

# Perform prediction

model <- train(isBound~ ., data = df, trControl = trainControl,

method = "glm", family = binomial, metric ="ROC")

summary(model)

## Plot AUROC

prediction <- prediction( model$pred$Y, model$pred$obs )

performance <- performance( prediction, "tpr", "fpr" )

plot(performance)

## Caluculate AUROC

auc <- performance(prediction, "auc")

auc <- unlist(slot(auc, "y.values"))

auc

**AUC scores (dependence of False positive rate versus True positive rate) for both logistic models for “1-mer” and “1mer+shape” features are almost the same (in the frames of the statistical error) that leads us to the thought that particulary in in vivo analysis for the CTCF transcription factor of Mys musculus data shape does not play any significant role- it either does not make worse nor improve the logistic regression model. Also, AUC≤≥0,8 demonstrates that our model is good for prediction DNA-binding speciricity.**

**LITERATURE.**

1. **Lecture of Tsu-Pei Chiu.**
2. **Beibei Xin and Tsu-Pei Chiu discussion cession.**
3. **T.P. Chiu, F. Comoglio, T. Zhou, L. Yang, R. Paro, and R. Rohs\*:**

**DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding.** [**Bioinformatics 32, 1211-1213 (2016)**](http://rohslab.cmb.usc.edu/Papers/2015_Chiu_DNAshapeR.pdf)